Handout 3 – Development Stages

1. Calibration

   Use the bisection method to find the root of $F(Y_0) = \phi(x = M \text{ using } \mathrm{Y}_0) - \phi_{\mathrm{RP}}$. Compute $\phi(M)$ using Runga-Kutta.

   a. testbisect.f: apply bisection to $F(Y_0) = Y_0^2 - 2$
   b. testrk4.f: apply RK4 to the ODE
   c. bracket.f: find bracketing interval for F
   d. getY0.f: the calibration code to solve (1). Outputs: Y_0, M_i, Y_i, H_i

   Issues that arose: what to use for the boundary condition at $x = L$. The first attempt, $\phi(L) = 0$, led to a boundary layer so we needed $dx = 1e - 5$ – too expensive. Changing the boundary condition allows us to use $dx = 1e - 3$, a hundred times faster.

2. Second Stage

   Replace the hard (3) with a simpler (linear) version to get the basic structure of the code. In the time loop, solve for phi in a subroutine. The first subroutine is for the linear version. No anticipated numerical problems. Code: fakephi.f

3. Third Stage

   Focus on (3). Solve the equation for one time step, assuming H_i^{n+1} is known. Begin with the most promising method. Test it carefully, because this is not a nice equation. Code: updatephi.f

4. Final Stage

   Put the pieces together. Copy fakephi.f, put in folder 1dmodel. Replace the phi subroutine with updatephi.f.